0.1 stepped pressure equilibrium code : mp00aa

1. Solves Beltrami linear system (for given helicity multiplier and poloidal flux), and returns interface rotational transform (via call to tr00ab).

2. This routine is only called by `ma02aa`, usually via `C05NBF`.

0.1.1 logical control

1. First, the matrix is assigned via a call to `ma01af` or `ma01ag`. This assigns the matrix quantities in `MR` (and `dmR`), and `MB`, only `if(Ldense)`, and `SCSF`, `irow` and `icol`, only `if(Lsparse)`. (These arrays were allocated in `ma02aa`.)

2. Different methods may be used to solve the linear system, i.e. to construct the Beltrami field in the given volume.

   - if `Ldense.eq.T` then
     - if `Lposdef=T` then : solution is provided by `F04ASF`, which assumes the matrix is symmetric positive-definite;
     - if `Lposdef=F` then : solution is provided by `F04ATF`;

     Both these routines require the matrix as an $N \times N$ array. (Note that `F04ASF` is faster than `F04ATF`; needs to be checked.)
   - The solution vector is 'unpacked' by `up00aa`. The unpacking routine must be consistent with the 'packing' description given in `global`.
   - if `Lsparse.eq.T` then will exploit the sparse structure of the matrix to reduce memory (and increase speed?). The routine `F11JEF` is used. Various parameters are given as input:
     - if `Lposdef=T` : use conjugate gradient, `method='CG'`
     - if `Lposdef=F` : use Lanczos method, `method='SYMMLQ'`
     - if `sparsepc=0` : no preconditionder, `precon='N'` (works terribly);
     - if `sparsepc=1` : Jacobi preconditionder, `precon='J'`
     - if `sparsepc=2` : SSOR preconditionder, `precon='S'`

     Other parameters, `sparsetol`,`sparseits` and `ssoromega`, are provided on input. The accuracy of the solution is controlled by `sparsetol`. (See the NAG documentation for details.) Note that if `sparseits.le.0`, then `sparseits=N` where `N` is the size of the matrix. Note that if `sparsetol.le.0.0`, then `tol=`ϵ where ϵ is machine precision. The initial guess for the iterative calculation is taken to be the last calculation solution. The sparse iterative approach needs more work to verify the speed, robustness and accuracy.
   - The sparse representation of the matrix, as determined by `ma01aa`, may contain repeated entries. These must be summed (by `F11ZBF`).

3. In the case `if(Ldense)` and `if(Lsparse)`, then the solutions will be compared (for debugging). In this case, the solution provided by `F11JEF` will be used for subsequent calculations.

4. Given the Beltrami field, the transform on each of the adjacent interfaces is computed by `tr00ab`, Note that `tr00ab` requires the radial derivatives of the vector potential to compute the magnetic field and the interface transforms, so `tr00ab` is only called if `Nofe.gt.0`.

5. This routine returns an 'error'-function, given as the difference between the computed transform of the constructed Beltrami field on each of the adjacent interfaces, ι, and the noble irrationals specified on input

$$\iota - \frac{p_l + \gamma p_r}{q_l + \gamma q_r}, \tag{1}$$

   where $p_l \equiv$`pl`, $q_l \equiv$`ql` are supplied on input and $\gamma = (1+\sqrt{5})/2$ is the golden mean.

6. This routine is typically called by `ma02aa` either directly, or indirectly via the NAG routine `C05NBF` which will adjust the Lagrange multiplier μ and the poloidal flux within each annulus ψ_p to set this error function to zero.

mp00aa.h last modified on 2011-10-11 ;